

PROJECT TITLE

College Name: Karuppannan Mariappan College

College Code: Bru3s

TEAM ID: NM2025TMID28582

TEAM MEMBERS: 4

Team LeaderName: YOGESWARAN K

Email: lokiofficial2004@gmail.com

Team Member1: MAYAVAN S

Email: mmayavan40@gmail.com

Team Member: HARINI S

Email: harinisekar0037@gmail.com

Team Member: KIRUTHIKA M

Email: kiruthikm4@gmail.com

1.INTRODUCTION

1.1 Project Overview

The Lease Management System is a Salesforce-based application designed to streamline the processes associated with leasing real estate properties. It handles tenant management, lease contracts, payments, and communication with automation features such as flows, approval processes, and email alerts.



1.2 Purpose

The main objective of the project is to enable organizations to efficiently manage properties, tenants, and lease-related activities. It reduces manual intervention, improves accuracy, and ensures better compliance and communication.

DEVELOPMENT PHASE

Creating Developer Account:

By using this URL - <https://www.salesforce.com/form/developer-signup/?d=pb>

Build enterprise-quality apps fast and get hands-on with Agentforce and Data Cloud.

Sign up for your Developer Edition.

- ✓ Build apps fast with drag-and-drop tools
- ✓ Go further with Apex code
- ✓ Build AI agents with Agentforce
- ✓ Harmonize your data with Data Cloud
- ✓ Ground Agentforce with structured and unstructured data
- ✓ Integrate with anything using APIs

Sign up for your Developer Edition

A free Salesforce Platform environment with Agentforce and Data Cloud

First name ✓ Last name ✓

Job title ✓ Work email

Company ✓ Country/Region ✓

Your org may be provisioned on or migrated to Hyperforce, Salesforce's public cloud infrastructure.

I agree to the Main Services Agreement – Developer Services and Salesforce Program Agreement. I acknowledge, as described in the Developer Documentation: (1) the Developer Edition includes autonomous and other generative AI features; and (2) Salesforce may limit use of those features and the org, and may terminate any org that has been inactive for 45 days.

We value your privacy. To learn more, visit our [Privacy Statement](#).

• Account Activation

Welcome to your Developer Edition

Hi Yogeswaran,

Thanks for signing up for a Developer Edition. Now you can start building on Salesforce for free and get hands-on with Agentforce and Data Cloud.

There's just one more step. Use the following link to reset the password for your Developer Edition. This link expires in 24 hours.

Reset Password

To easily log in later, save this URL:
<https://orgfarm-3192e294bf-dev-ed.develop.my.salesforce.com>

- Created objects: Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Object Manager interface for the 'property' object. The left sidebar lists various configuration tabs: Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules. The main 'Details' tab is selected. The 'Description' section contains fields for API Name (property__c), Singular Label (property), and Plural Label (properties). On the right, deployment status is set to 'Deployed' with 'Help Settings' and a link to the 'Standard salesforce.com Help Window'. Navigation icons for back, forward, search, and refresh are at the top.

The screenshot shows the Salesforce Object Manager interface for the 'Tenant' object. The left sidebar lists the same configuration tabs as the previous screenshot. The main 'Details' tab is selected. The 'Description' section contains fields for API Name (Tenant__c), Singular Label (Tenant), and Plural Label (Tenants). On the right, deployment status is set to 'Deployed' with 'Help Settings' and a link to the 'Standard salesforce.com Help Window'. Navigation icons for back, forward, search, and refresh are at the top.

SETUP > OBJECT MANAGER

lease

Details

Description

API Name: lease_c
Custom: ✓
Singular Label: lease
Plural Label: lease

Enable Reports
✓
Track Activities
✓
Track Field History
✓
Deployment Status: Deployed
Help Settings: Standard salesforce.com Help Window

Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules

Edit **Delete**

SETUP > OBJECT MANAGER

Payment for tenant

Details

Description

API Name: Payment_for_tenant_c
Custom: ✓
Singular Label: Payment for tenant
Plural Label: Payment

Enable Reports
✓
Track Activities
✓
Track Field History
✓
Deployment Status: Deployed
Help Settings: Standard salesforce.com Help Window

Fields & Relationships
Page Layouts
Lightning Record Pages
Buttons, Links, and Actions
Compact Layouts
Field Sets
Object Limits
Record Types
Related Lookup Filters
Search Layouts
List View Button Layout
Restriction Rules
Scoping Rules

Edit **Delete**

- Configured fields and relationships

Fields & Relationships
9 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Address	Address__c	Long Text Area(32768)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Name	Name__c	Text(25)		
Owner	OwnerId	Lookup(User,Group)		✓
property	property__c	Lookup(property)		✓
property Name	Name	Text(80)		✓
sfqt	sfqt__c	Text(18)		
Type	Type__c	Picklist		

Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Amount	Amount__c	Number(18, 0)		
check-for-payment	check_for_payment__c	Picklist		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Payment date	Payment_date__c	Date		
Payment Name	Name	Text(80)		✓

Setup Home Object Manager

SETUP > OBJECT MANAGER
lease

Fields & Relationships 7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
End date	End_date_c	Date		
Last Modified By	LastModifiedById	Lookup(User)		
lease Name	Name	Text(80)		✓
Owner	OwnerId	Lookup(User/Group)		✓
property	property_c	Lookup(property)		✓
start date	start_date_c	Date		

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Restriction Rules Scoping Rules

Setup Home Object Manager

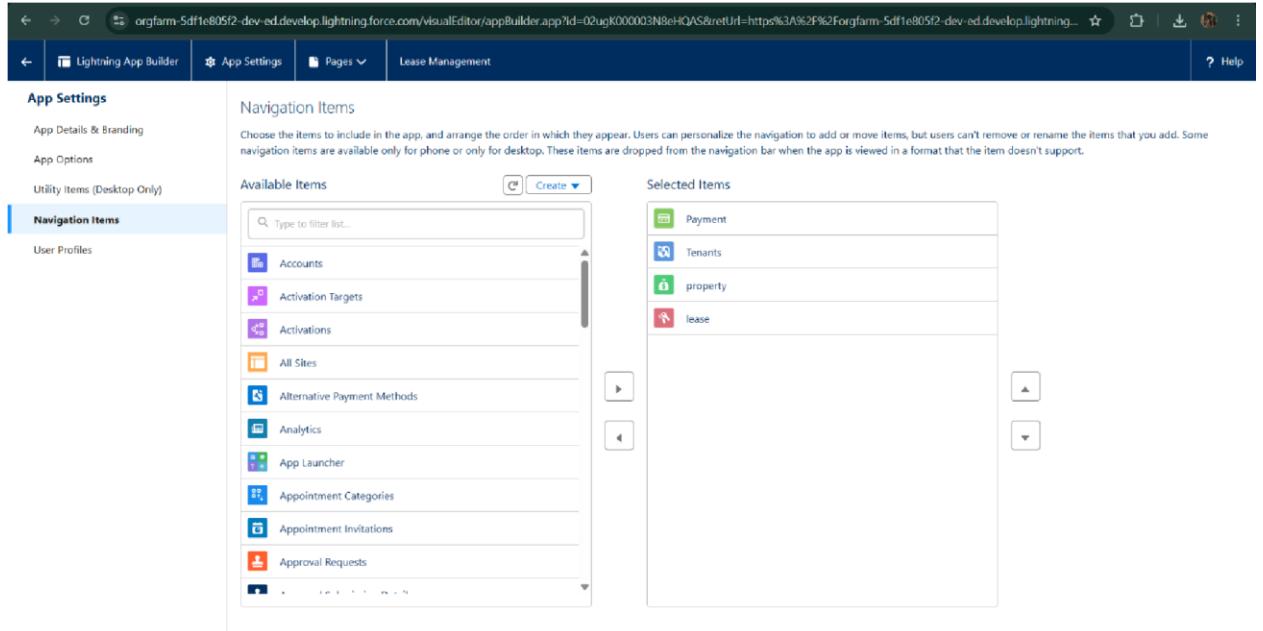
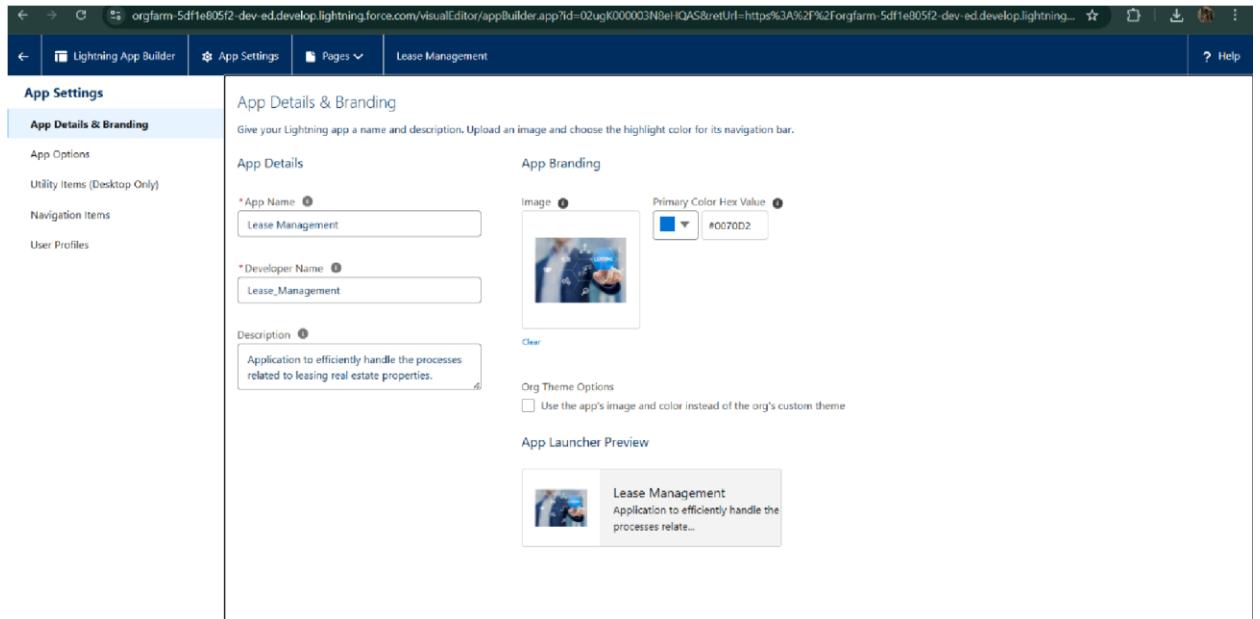
SETUP > OBJECT MANAGER
Tenant

Fields & Relationships 7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email	Email_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User/Group)		✓
Phone	Phone_c	Phone		
status	status_c	Picklist		
Tenant Name	Name	Name		✓

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Restriction Rules Scoping Rules

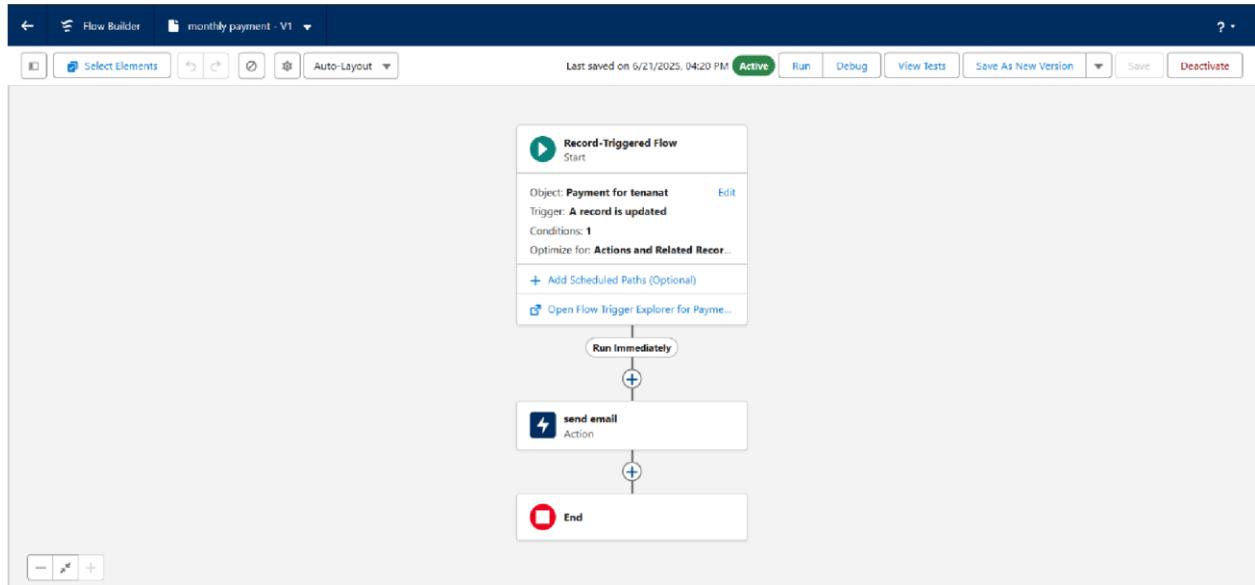
- Developed Lightning App with relevant tabs



The screenshot shows the 'User Profiles' section of the Lightning App Builder. On the left, there's a sidebar with 'App Settings' and 'Navigation Items'. Under 'App Settings', 'User Profiles' is selected. The main area has two sections: 'Available Profiles' on the left and 'Selected Profiles' on the right. The 'Available Profiles' list includes: Analytics Cloud Integration User, Analytics Cloud Security User, Anypoint Integration, Authenticated Website, B2B Reordering Portal Buyer Profile, Contract Manager, Custom: Marketing Profile, Custom: Sales Profile, Custom: Support Profile, and Customer Community Login User. The 'Selected Profiles' list contains 'System Administrator'. A search bar at the top says 'Choose the user profiles that can access this app.'

The screenshot shows the 'Lease Management' interface. At the top, there are tabs for 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. Below the tabs is a search bar and a 'Recently Viewed' section. The 'Recently Viewed' section is titled 'Payment' and lists 5 items: 'Rahul', 'Jack', 'Raj', 'Sam', and 'Lahari'. To the right of this list is a table with columns for 'Payment Name' and other details, with checkboxes next to each row. The table has several rows, though only the first five are clearly visible.

- Implemented Flows for monthly rent and payment success



- To create a validation rule to a Lease Object

Validation Rule Edit

Rule Name: lease_end_date

Active:

Description:

Error Condition Formula:

Example: Discount_Percent_c>0.30 More Examples...

If this formula expression is true, display the text defined in the Error Message area.

Insert Field: Insert Operator:

Formula: End_date_c <= start_date_c

Functions:

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII
- ASIN

Quick Tips: Operators & Functions

Help on this function: ABS(*number*) Returns the absolute value of a number, a number without its sign.

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes links for Setup, Home, and Object Manager. The main title is "lease Validation Rule". The validation rule details are as follows:

- Rule Name: lease_end_date
- Error Condition Formula: End_date__c <= start_date__c
- Error Message: Your End date must be greater than start date
- Description: (empty)
- Created By: Sowmya Team, 6/19/2025, 5:37 AM
- Modified By: Sowmya Team, 6/26/2025, 7:47 AM

The sidebar on the left lists various configuration options for the lease object, such as Details, Fields & Relationships, Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, and Scoping Rules.

- Added Apex trigger to restrict multiple tenants per property

The screenshot shows the Salesforce Lease Management interface. A modal window titled "New Tenant" is open, displaying an "Information" section with fields for Tenant Name (chinnu) and Email (chinnu@gmail.com). An error message at the bottom states: "We hit a snag. Review the errors on this page. * A tenant can have only one property". The background shows a list of tenants with one item visible: "1 niranjan".

- Scheduled monthly reminder emails using Apex class

```

1 global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10        }
11    }
12
13 }
14
15
16 public static void sendMonthlyEmails() {
17
18     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20     for (Tenant__c tenant : tenants) {
21
22         String recipientEmail = tenant.Email__c;
23
24         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due. Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain';
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30         email.setToAddresses(new String[]{recipientEmail});
31
32         email.setSubject(emailSubject);
33
34         email.setPlainTextBody(emailContent);
35

```

- Built and tested email templates for leave request, approval, rejection, payment, and reminders

Search Setup

Classic Email Templates

Email Template Detail

Leave approved

Preview your email template below.

Email Template Name	Leave approved	Available For Use
Template Unique Name	Leave_approved	Last Used Date
Encoding	Unicode (UTF-8)	Times Used
Author	Sowmya Team (Change)	
Description		
Created By	Sowmya Team	Modified By
	6/20/2025, 1:08 AM	Sowmya Team

Email Template

Plain Text Preview

dear{!Tenant__c.Name},

I hope this message finds you well. I am writing to inform you that I have received your email confirming the approval of my leave request. I would like to express my gratitude for considering and approving my time off.

The screenshot shows the Salesforce Setup interface. In the top left, there's a blue cloud icon. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Setup' with a magnifying glass icon. On the far right of the top bar are several small icons: a star, a plus sign, a house, a question mark, a gear, a red circle with a number '2', and a user profile.

The main content area has a title 'SETUP Classic Email Templates'. Below it, a section titled 'Text Email Template' shows a template named 'tenant leaving'. The preview pane displays the email content:

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	tenant leaving
Template Unique Name	tenant_leaving
Encoding	Unicode (UTF-8)
Author	Sowmya_Team [Change]
Description	
Created By	Sowmya_Team 6/20/2025, 1:06 AM
Modified By	Sowmya_Team 6/20/2025, 1:06 AM

Email Template

Subject: request for approve the leave

Plain Text Preview

```
Dear {Tenant__c.CreatedBy},  
Please approve my leave
```

This screenshot shows the same Salesforce Setup interface as the first one, but with a different email template.

The main content area has a title 'SETUP Classic Email Templates'. Below it, a section titled 'Text Email Template' shows a template named 'Leave rejected'. The preview pane displays the email content:

Email Template Detail

Email Templates from Salesforce	Unified Public Classic Email Templates
Email Template Name	Leave rejected
Template Unique Name	Leave_rejected
Encoding	Unicode (UTF-8)
Author	Sowmya_Team [Change]
Description	
Created By	Sowmya_Team 6/20/2025, 1:11 AM
Modified By	Sowmya_Team 6/20/2025, 1:11 AM

Email Template

Subject: Leave rejected

Plain Text Preview

```
Dear {Tenant__c.Name},  
I hope this email finds you well. Your contract has not ended. So we can't approve your leave  
your leave has rejected
```

Classic Email Templates

Tenant Email

Email Template Detail

- Email Template Name: Tenant Email
- Template Unique Name: tenant_email
- Encoding: Unicode (UTF-8)
- Author: Scamya Team [Change]
- Description: Created by Scamya Team, 6/20/2025, 1:12 AM
- Modified By: Scamya Team, 6/20/2025, 1:12 AM

Email Template

Plain Text Preview

Dear {tenant_c_name},

I trust this email finds you well. We appreciate your continued tenancy at our property and I hope you have been comfortable in your residence.

Classic Email Templates

Tenant payment

Email Template Detail

- Email Template Name: tenant_payment
- Template Unique Name: tenant_payment
- Encoding: Unicode (UTF-8)
- Author: Scamya Team [Change]
- Description: Created by Scamya Team, 6/20/2025, 1:13 AM
- Modified By: Scamya Team, 6/20/2025, 1:13 AM

Email Template

Plain Text Preview

Dear {tenant_c_email_c},

We hope this email finds you well. We are writing to inform you that we have successfully received your monthly payment. Thank you for your prompt and diligent payment.

- Approval Process creation

For Tenant Leaving:

Process Definition Detail

Process Name	TenantApproval	Active	
Unique Name	TenantApproval		
Description			
Entry Criteria	tenant__r.status EQUALS Stay		
Record Editability	Administrator ONLY		
Approval Assignment Email Template		Allow Submitters to Recall Approval Requests	
Initial Submitters	Tenant Owner		
Created By	Sowmya_Team, 6/23/2025, 3:41 AM	Modified By	Sowmya_Team, 6/26/2025, 11:57 PM

Initial Submission Actions

Action	Type	Description
	Record Lock	Lock the record from being edited

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	Step 1			User:Sowmya_Team	Final Rejection

For Check for Vacant:

Process Definition Detail

Process Name	check for vacant	Active	
Unique Name	check_for_vacant		
Description			
Entry Criteria	Tenant: status EQUALS Leaving		
Record Editability	Administrator ONLY		
Approval Assignment Email Template	leave_approved	Allow Submitters to Recall Approval Requests	
Initial Submitters	Tenant Owner		
Created By	Sowmya_Team, 6/20/2025, 3:18 AM	Modified By	Sowmya_Team, 6/26/2025, 11:02 PM

Initial Submission Actions

Action	Type	Description
Edit Remove	Record Lock	Lock the record from being edited
	Email Alert	RECALL_APPROVALS.MY.JAVA

Approval Steps

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions Edit	1	step1			User:Sowmya_Team	Final Rejection

- Apex Trigger

Create an Apex Trigger

Screenshot of the Salesforce IDE showing the creation of an Apex Handler class.

The top window shows the code editor for `test.apex` with the following trigger definition:

```

1 trigger test on Tenant__c (before insert)
2 {
3     if(trigger.isInsert && trigger.isBefore){
4         testHandler.preventInsert(trigger.new);
5     }
6 }

```

A modal dialog titled "Open" is displayed, listing entities for selection. The "Triggers" option is selected, and the entry "test" is visible in the list.

The bottom window shows the browser developer console with the URL `orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage`. It displays the same Apex code as the IDE.

Create an Apex Handler class

Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```

File Edit Debug Test Workspace Help < >
testHandler.apc MonthlyEmailScheduler.apc
Code Coverage: None API Version: 64 Go To
1 * public class testHandler {
2
3 *   public static void preventInsert(List<Tenant__c> newList) {
4
5     Set<Id> existingPropertyIds = new Set<Id>();
6
7     for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9       existingPropertyIds.add(existingTenant.Id);
10
11     }
12
13     for (Tenant__c newTenant : newList) {
14
15       if (newTenant.Property__c != null) {
16
17         newTenantaddError('A tenant can have only one property');
18
19       }
20
21     }
22
23   }
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Entity Type

Entry Type	Name	Namespace	Related
Classes	testHandler	MonthlyEmailScheduler	↪ test ApexTrigger ↪ property CustomField ↪ Tenant__c SObject ↪ Tenant__c SObject
Triggers			
Pages			
Components			
Objects			
Static Resources			
Package			

Open Filter Hide Managed Packages Refresh

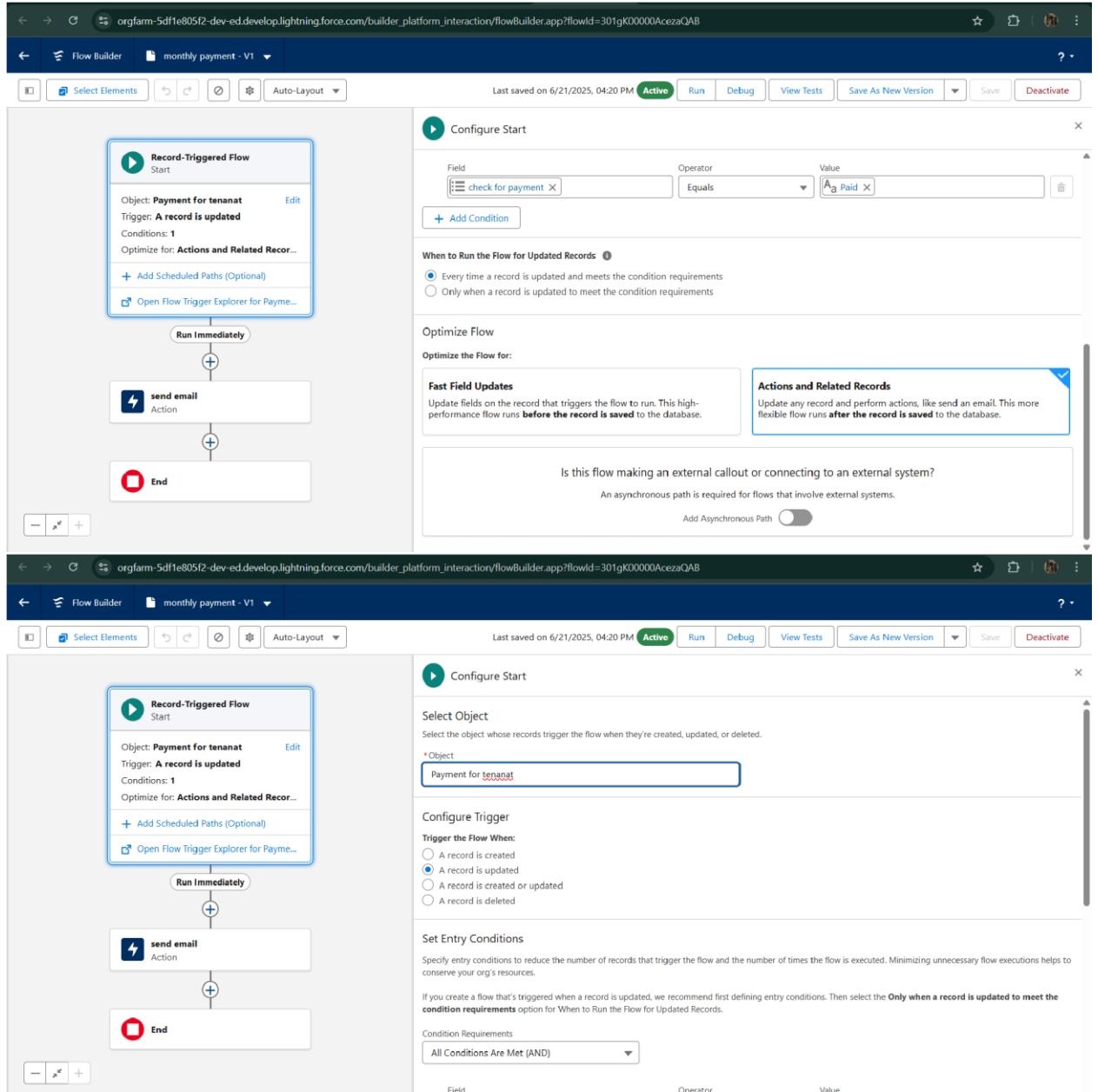
Developer Console - Google Chrome
 orgfarm-5df1e805f2-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

```

File Edit Debug Test Workspace Help < >
testHandler.apc MonthlyEmailScheduler.apc
Code Coverage: None API Version: 64 Go To
1 * public class testHandler {
2
3 *   public static void preventInsert(List<Tenant__c> newList) {
4
5     Set<Id> existingPropertyIds = new Set<Id>();
6
7     for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE Property__c != null]) {
8
9       existingPropertyIds.add(existingTenant.Property__c);
10
11     }
12
13     for (Tenant__c newTenant : newList) {
14
15       if (newTenant.Property__c != null && existingPropertyIds.contains(newTenant.Property__c)) {
16
17         newTenantaddError('A tenant can have only one property');
18
19       }
20
21     }
22
23   }
  
```

Logs Tests Checkpoints Query Editor View State Progress Problems

- FLOWS



- Schedule class:
Create an Apex Class

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apex [] testHandler.apex [] MonthlyEmailScheduler.apex []

Code Coverage: None API Version: 54 Go To

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10        }
11    }
12
13 }
14
15
16 public static void sendMonthlyEmail
17 {
18     List<Tenant__c> tenants = [SELECT
19
20         for (Tenant__c tenant : tenants
21
22             String recipientEmail = tenant.Email__c;
23

```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Open Entity Type Entities Namespace Related

Entry Type	Name	Namespace	Name	Extent	Direction
Classes	testHandler		MonthlyEmailScheduler	Email	CustomField References
Triggers				← Tenant__c	Object References
Pages				← Tenant__c	Object References
Page Components					
Objects					
Static Resources					
Packages					

Open Filter Filter the repository (* = any string) Hide Managed Packages Refresh

Developer Console - Google Chrome

File Edit Debug Test Workspace Help < >

test.apex [] testHandler.apex [] MonthlyEmailScheduler.apex []

Code Coverage: None API Version: 54 Reset

```

1 * global class MonthlyEmailScheduler implements Schedulable {
2
3     global void execute(SchedulableContext sc) {
4
5         Integer currentDay = Date.today().day();
6
7         if (currentDay == 1) {
8
9             sendMonthlyEmails();
10        }
11    }
12
13 }
14
15
16 public static void sendMonthlyEmails()
17 {
18     List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
19
20     for (Tenant__c tenant : tenants)
21
22         String recipientEmail = tenant.Email__c;
23
24         String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due now. Timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';
25
26         String emailSubject = 'Reminder: Monthly Rent Payment Due';
27
28         Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
29
30         email.setPlainText(emailContent);
31
32         Messaging.SingleEmailMessage[] emails = new Messaging.SingleEmailMessage[1];
33
34         emails[0] = email;
35
36         Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});
37
38
39     }
40
41 }
42

```

Logs Tests Checkpoints Query Editor View Status Progress Problems

Schedule Apex class

The screenshot shows the Salesforce Setup Apex Classes page. The left sidebar has a search bar and navigation links for Apex Exception Email, Custom Code (Apex Classes selected), Apex Settings, Apex Test Execution, Apex Test History, Apex Triggers, Environments, Jobs, Apex Flex Queue, and Apex Jobs. A note says "Didn't find what you're looking for? Try using Global Search." The main area displays the Apex Class MonthlyEmailScheduler. The Apex Class Detail section shows the Name as MonthlyEmailScheduler, Namespace Prefix as None, Created By as Sowmya Team, and Last Modified By as Sowmya Team. The Status is Active, Code Coverage is 0% (0/15), and Last Modified Date is 6/23/2025, 2:47 AM. Below this is the Class Body tab, which contains the following Apex code:

```
1 global class MonthlyEmailScheduler implements Scheduleable {
2     global void execute(SchedulableContext sc) {
3         Integer currentDay = Date.today().day();
4         if (currentDay == 1) {
5             sendMonthlyEmails();
6         }
7     }
8     public static void sendMonthlyEmails() {
9         List<Tenant__c> tenants = [SELECT Id, Email__c FROM Tenant__c];
10        for (Tenant__c tenant : tenants) {
11            ...
12        }
13    }
14}
15
16
17
18
19
20
```

The screenshot shows the Salesforce Lease Management Tenant record details page for Aswini. The left sidebar has links for Lease Management, Payment, Tenants (selected), property, and lease. The main area shows the Tenant Aswini record with tabs for Related and Details. The Details tab displays fields: Tenant Name (Aswini), Owner (Sowmya Team), Email (aswinimaraadi15@gmail.com), Phone ((905) 223-5567), status (Leaving), property (Imran), Created By (Sowmya Team, 6/26/2025, 6:05 AM), and Last Modified By (Sowmya Team, 6/26/2025, 11:06 PM). At the bottom are Cancel and Save buttons. On the right, there is an Activity sidebar with options like New Case, New Lead, Delete, Clone, Change Owner, Refresh, and Submit for Approval. It also shows sections for Upcoming & Overage and a note about past activity.

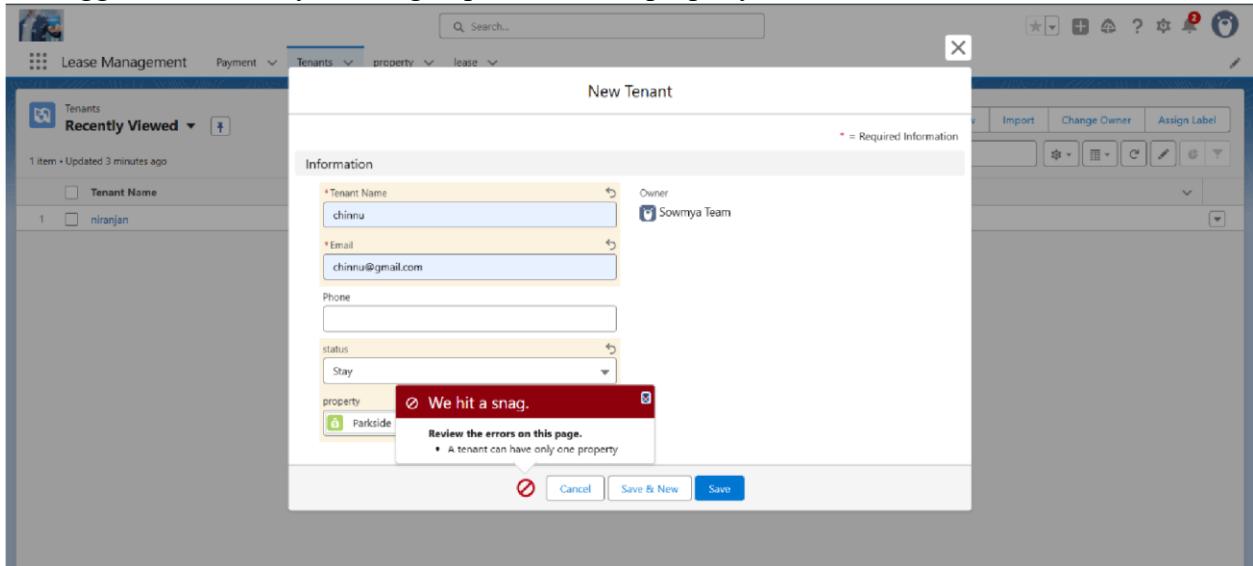
The screenshot shows a Salesforce Lightning page for 'Lease Management'. The top navigation bar includes tabs for 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. A green banner at the top right indicates 'Tenant was submitted for approval.' The main form contains fields for 'Tenant Name' (Aswini), 'Email' (aswiniamaraadi15@gmail.com), 'Phone' ((905) 223-5567), 'status' (Leaving), and 'property' (Imran). The 'Owner' field is set to 'Sowmya Team'. Below the form, it shows 'Created By' (Sowmya Team) and 'Last Modified By' (Sowmya Team). On the right side, there is an 'Activity' sidebar with sections for 'Upcoming & Overdue' and 'Past activity'. The status message 'No activities to show.' is displayed. The bottom of the screen shows a Windows taskbar with various icons and the date/time (27-06-2025, 12:46).

The screenshot shows a Salesforce Lightning page for a 'Process Instance Step' titled 'Tenant Approval'. The status is 'Approved'. The details section shows the 'Submitter' as 'Sowmya Team', 'Date Submitted' as 'Jun 27, 2025', 'Actual Approver' as 'Sowmya Team', and 'Assigned To' as 'Sowmya Team'. On the right, a 'Notifications' sidebar lists five notifications all from 'Aswini': 'Approval request for the tenant is approved' (a few seconds ago), 'Approval request for the tenant is rejected' (an hour ago), 'Approval request for the tenant is approved' (an hour ago), 'Approval request for the tenant is approved' (an hour ago), and 'Approval request for the tenant is approved' (19 hours ago). The bottom of the screen shows a Windows taskbar with various icons and the date/time (27-06-2025, 12:46).

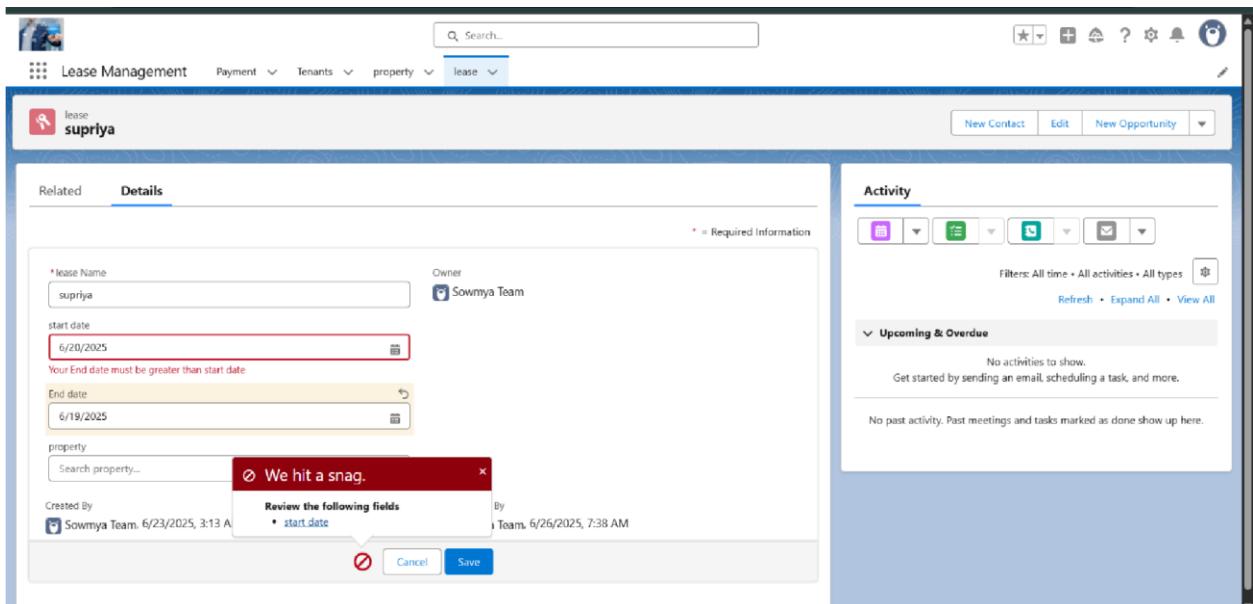
FUNCTIONAL AND PERFORMANCE TESTING

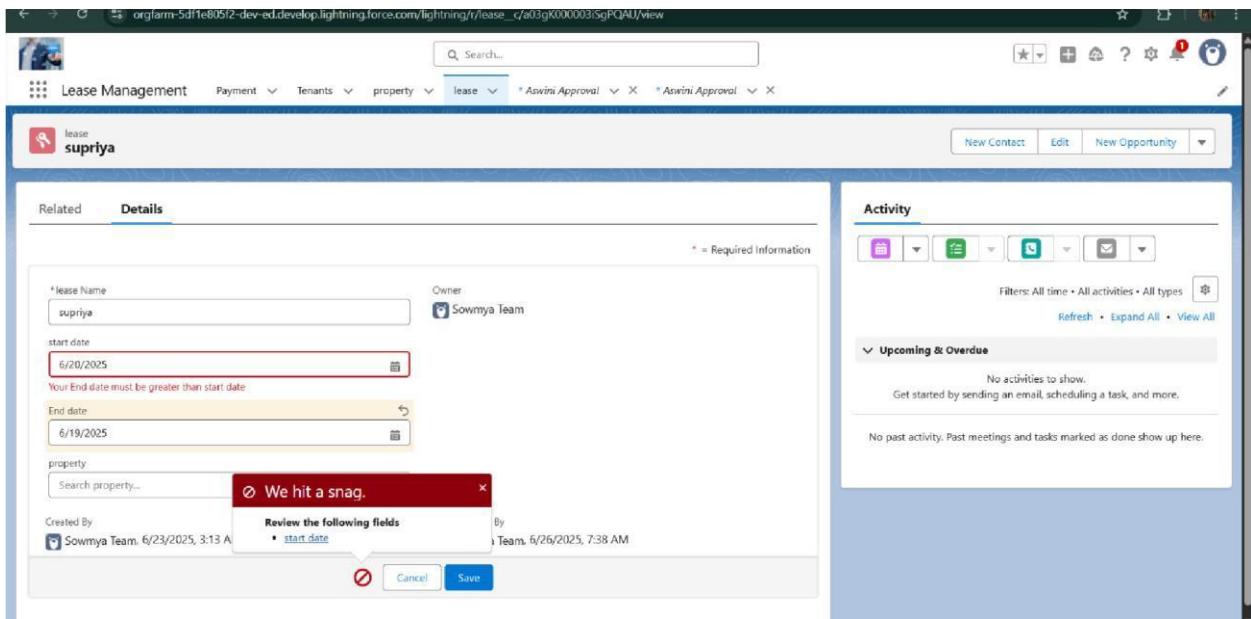
Performance Testing

- Trigger validation by entering duplicate tenant-property records

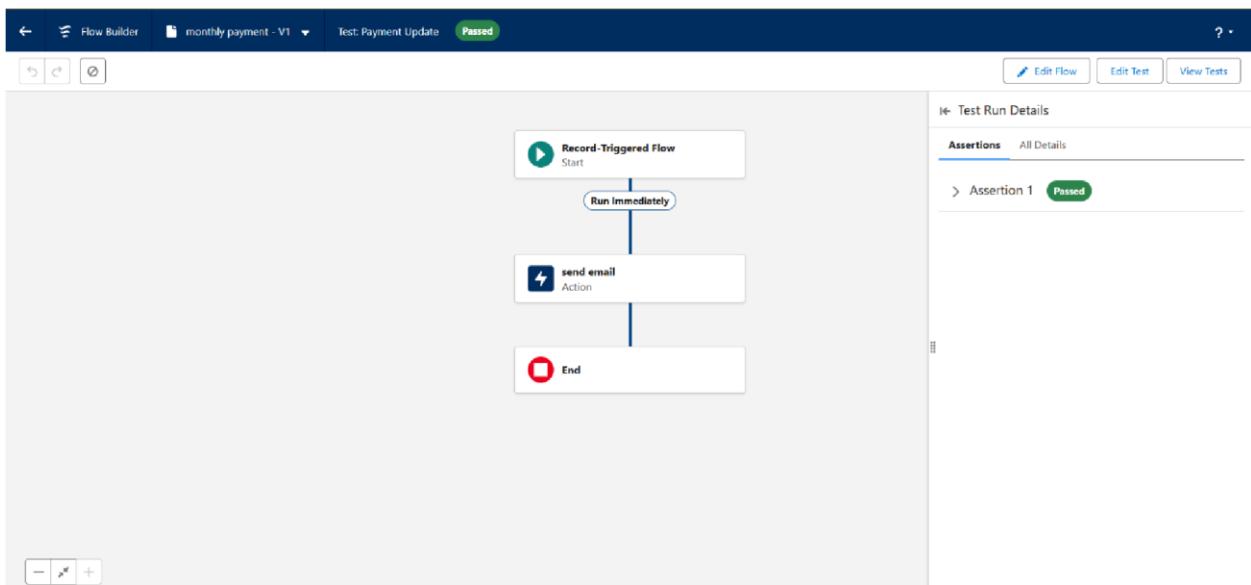


- Validation Rule checking





- Test flows on payment update



- Approval process validated through email alerts and status updates

Tenant: niranjan

Related Details

* = Required Information

Tenant Name niranjan	Owner Sowmya Team
Email niranjan1506@gmail.com	
Phone	
status Stay	
property Parksid Lofts	

Created By
Sowmya Team, 6/23/2025, 2:33 AM

Last Modified By
Sowmya Team, 6/23/2025, 3:58 AM

Cancel **Save**

Notifications

- Approval request for the tenant is approved niranjan (a few seconds ago)
- Approval request for the tenant is rejected niranjan (Jun 23, 2025, 4:29 PM)
- Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:25 PM)
- Approval request for the tenant is approved niranjan (Jun 23, 2025, 4:14 PM)
- New Guidance Center learning resource available Define Your Sales Process Learn how to guide reps through the sales process. (Jun 20, 2025, 1:28 PM)

Tenant: niranjan

Approval History (6+)

Step Name	Date	Status	Assigned To
Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team
Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team
Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team
Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team
Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team

Payment (2)

Payment Name
Jack
Rahul

New Contact **Edit** **New Opportunity**

No past activity. Past meetings and tasks marked as done show up here.

RESULTS

Output Screenshots

- Tabs for Property, Tenant, Lease, Payment

The screenshot shows the Salesforce Setup interface under the 'Tabs' section. It displays a table of custom object tabs with columns for Action, Label, Tab Style, and Description. The tabs listed are:

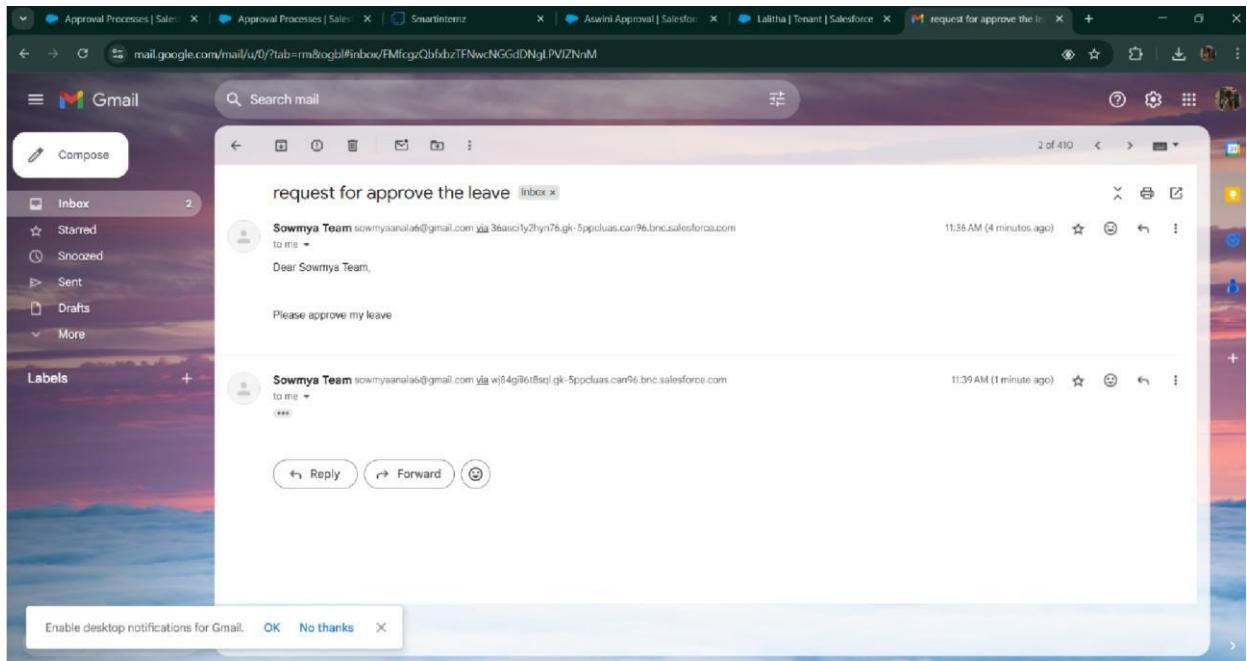
Action	Label	Tab Style	Description
Edit Del	lease	Key	
Edit Del	Payment	Credit card	
Edit Del	property	Back	
Edit Del	Tenants	Map	

- Email alerts

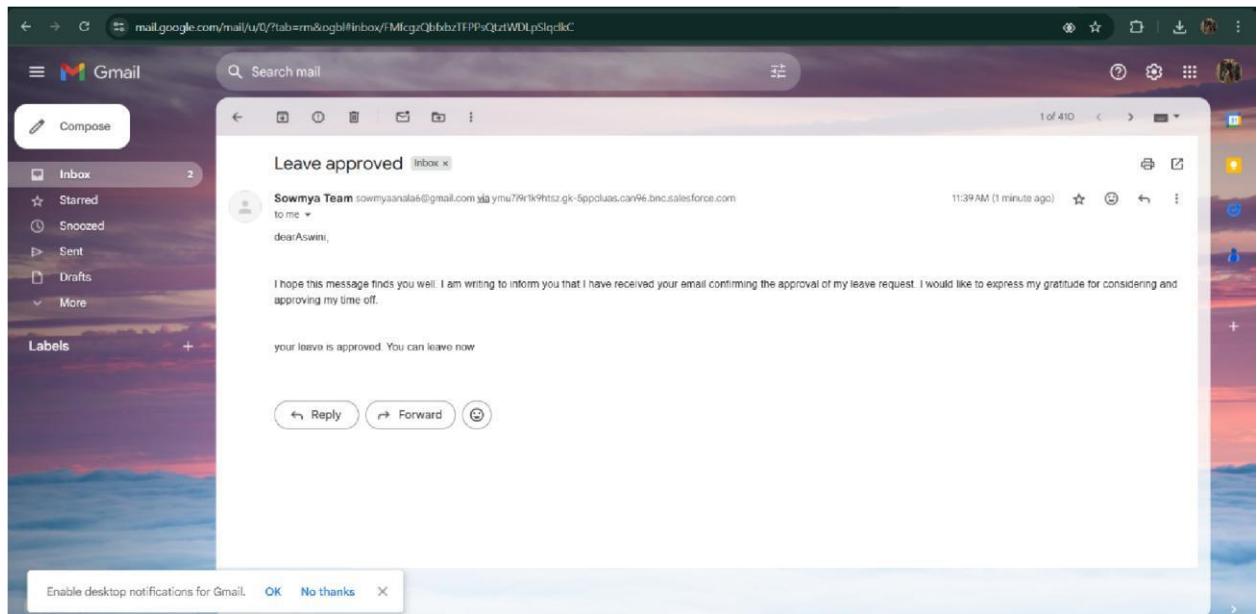
The screenshot shows the 'Lease Management' application interface. The top navigation bar includes 'Lease Management', 'Payment', 'Tenants', 'property', and 'lease'. Below the navigation is a search bar and a toolbar with various icons. The main area is titled 'Approval History' and shows a table with 8 items. The table columns are: Step Name, Date, Status, Assigned To, Actual Approver, and Comments. The data in the table is as follows:

Step Name	Date	Status	Assigned To	Actual Approver	Comments
1 Step 1	6/25/2025, 5:39 AM	Approved	Sowmya Team	Sowmya Team	approved
2 Approval Request Submitted	6/25/2025, 5:39 AM	Submitted	Sowmya Team	Sowmya Team	leaving
3 Step 1	6/23/2025, 3:59 AM	Rejected	Sowmya Team	Sowmya Team	Rejected
4 Approval Request Submitted	6/23/2025, 3:58 AM	Submitted	Sowmya Team	Sowmya Team	Leaving
5 Step 1	6/23/2025, 3:55 AM	Approved	Sowmya Team	Sowmya Team	Approved
6 Approval Request Submitted	6/23/2025, 3:55 AM	Submitted	Sowmya Team	Sowmya Team	leaving
7 Step 1	6/23/2025, 3:44 AM	Approved	Sowmya Team	Sowmya Team	Approval Approved
8 Approval Request Submitted	6/23/2025, 3:42 AM	Submitted	Sowmya Team	Sowmya Team	Leaving

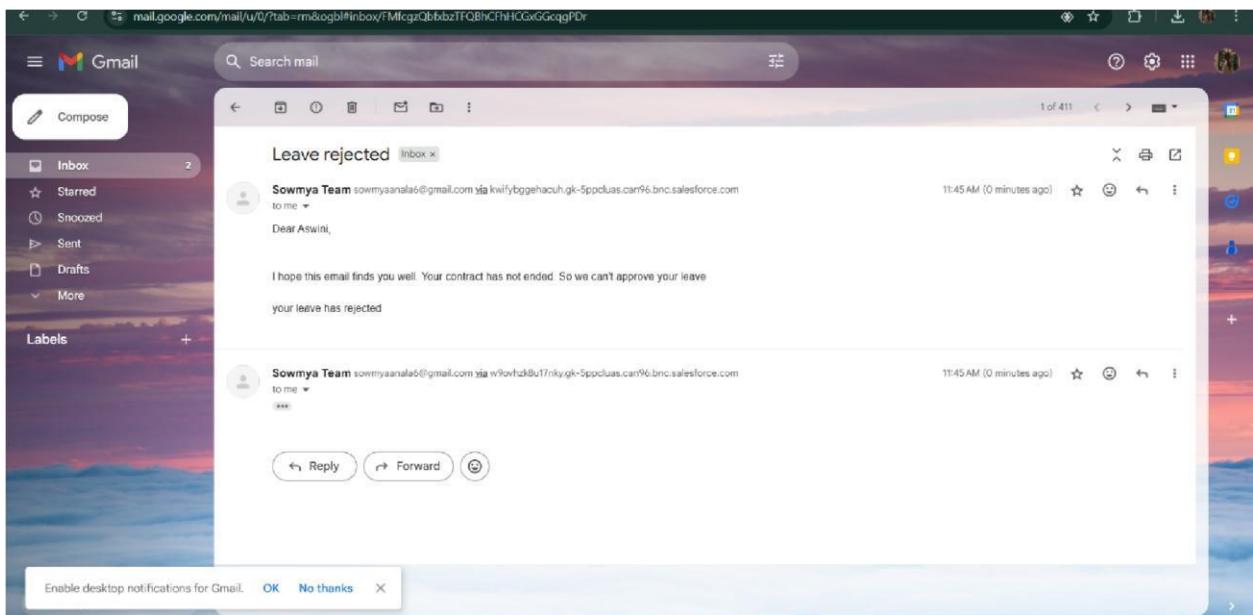
- Request for approve the leave



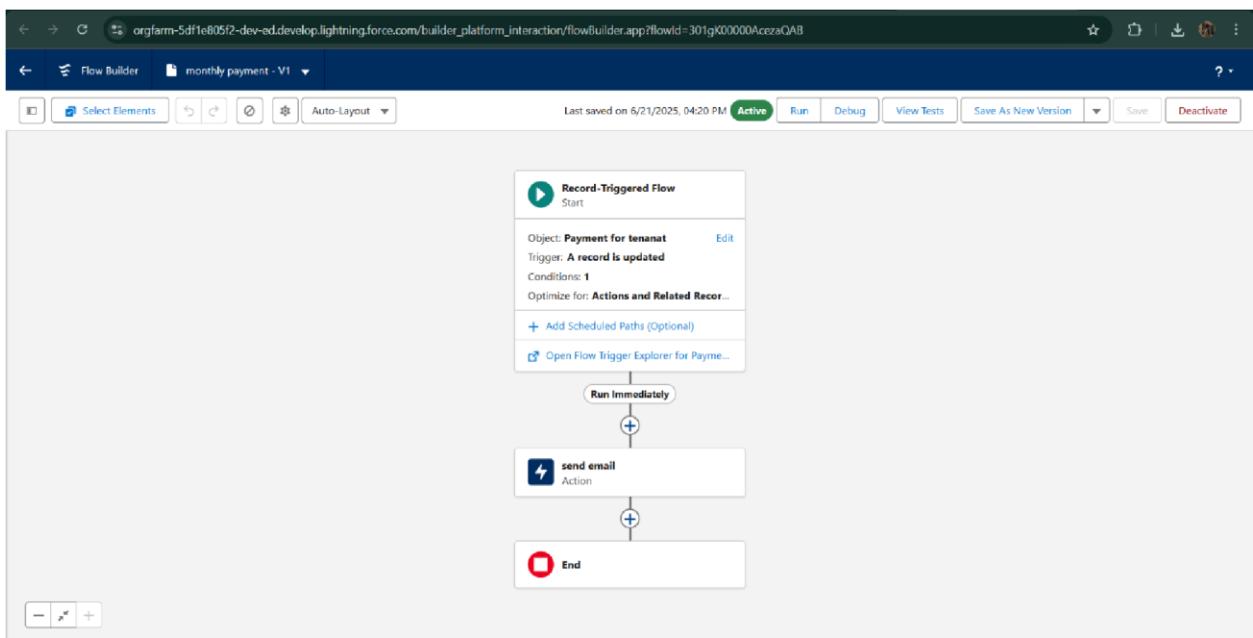
- Leave approved



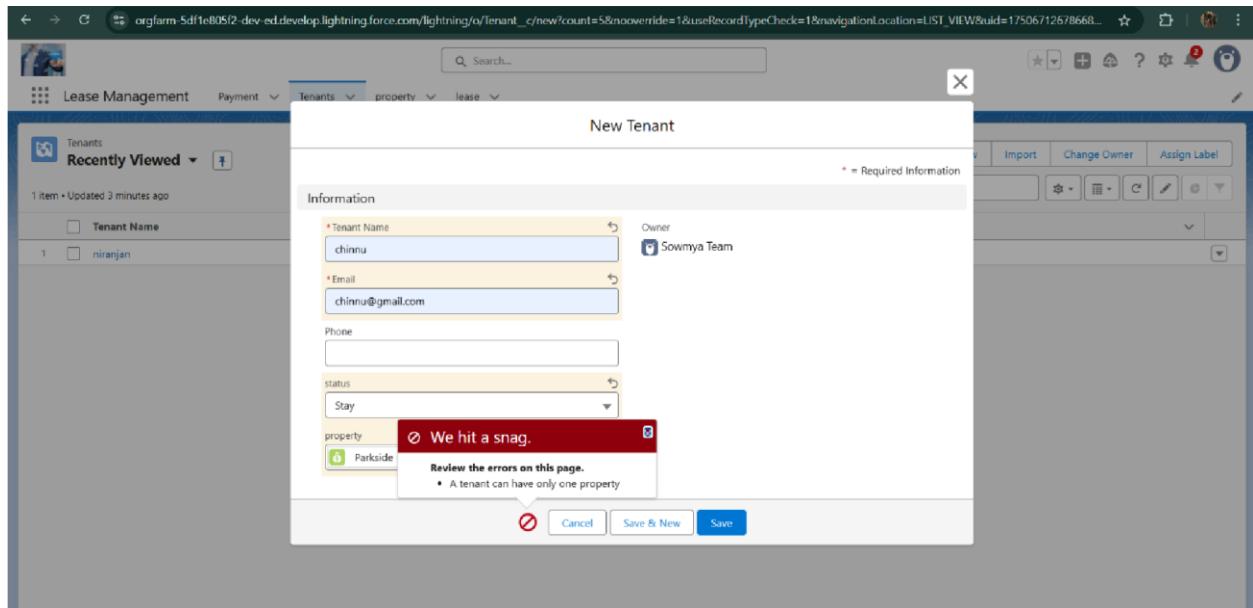
- Leave rejected



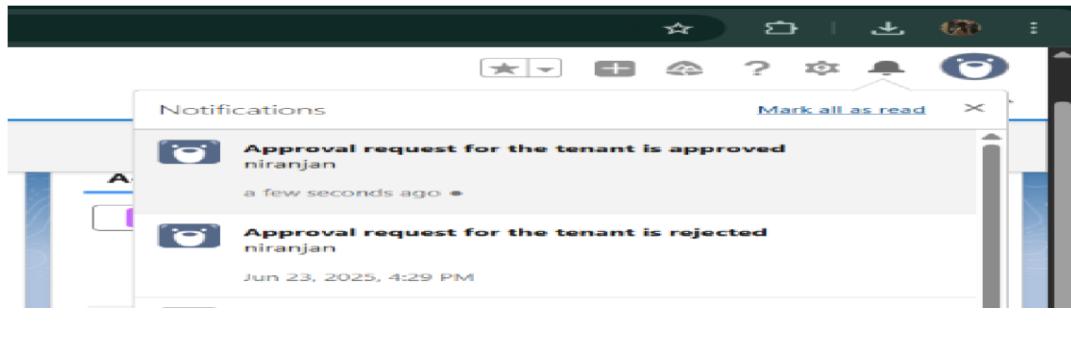
- Flow runs



- Trigger error messages



- Approval process notifications



ADVANTAGES & DISADVANTAGES

CONCLUSION

The Lease Management System successfully streamlines the operations of leasing through a structured, automated Salesforce application. It improves efficiency, communication, and data accuracy for both admins and tenants.

APPENDIX

- **Source Code:** Provided in Apex Classes and Triggers

Test.apxt: trigger test on Tenant__c

```
(before insert) { if(trigger.isInsert &&
trigger.isBefore){
testHandler.preventInsert(trigger.new);
```

} } **testHandler.apxc:**

```
public class
```

```
testHandler { public
```

```
static void
```

```
preventInsert(List<
```

```
Tenant__c> newList)
```

```
{ Set<Id>
```

```
existingPropertyIds
```

```
= new Set<Id>()
```

```
for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c
WHERE Property__c != null]) {
```

```
existingPropertyIds.add(existingTenant.Property__c;
```

```
} for (Tenant__c newTenant :
```

```

        newlist) {

            if (newTenant.Property__c != null &&
existingPropertyIds.contains(newTenant.Property__c)) { newTenant.addError('A
tenant can have only one property');

        }

    }

}

```

MothlyEmailScheduler.apxc:

```

global class MonthlyEmailScheduler implements Schedulable {
    global
        void execute(SchedulableContext sc) {
            Integer currentDay =
Date.today().day();
            if (currentDay == 1) {
                sendMonthlyEmails();
            }
        }
    public static void
        sendMonthlyEmails() {
            List<Tenant__c>
tenants = [SELECT Id, Email__c FROM
Tenant__c];
            for (Tenant__c tenant :
tenants) {
                String recipientEmail = tenant.Email__c;
                String emailContent =
'I trust this email finds you well. I am writing to remind you
that the monthly rent is due Your timely payment ensures the smooth functioning of our
rental arrangement and helps maintain a positive living environment for all.';
                String emailSubject = 'Reminder: Monthly Rent Payment Due';
                Messaging.SingleEmailMessage email = new

```

```
        Messaging.SingleEmailMessage(); email.setToAddresses(new
String[]{recipientEmail}); email.setSubject(emailSubject);
email.setPlainTextBody(emailContent);

        Messaging.sendEmail(new Messaging.SingleEmailMessage[]{email});

    }

}
```